รายการอ้างอิง

Collin, R. E., *The Fundamental of Microwave Engineering*., U.S.A : Mcgraw Hill, 1966.

Forsythe, W. D., *Smithsonian Physical Tables*., The Smithsonian Institution, 1964.

Jackson, J. D., *Classical Electrodynamics*., 2nd. ed. U.S.A. : John Wiley, 1966.

Jenkins, T. E., *Semiconductor Science Growth and Characterization Techniques*., U.K. : Prentice Hall, 1995.

Kernighan, J., Ritchie, D., *C Programing Language*., U.S.A. : Prentice Hall, 1991.

Montgomery, C. G., *Technique of Microwave Measurement*., U.S.A. : Dover, 1966

Pain, H. J., *The Physics of Vibrations and Wave*., 2nd.ed., U.K. : John Wiley & Sons, 1978.

Reitz, J. R., Milford, F. J., Christy, R. W., *Foundations of Electromagnetic Theory*., 4 th.ed., U.S.A : Addision Wesley, 1992.

Subramanian, V., Sobhanadri, J., *Rev. Sci. Instrum*., Vol. 64, 1993.

Subramanian, V., Sobhanadri, J., *Rev. Sci. Instrum*., Vol. 65(2), 1994.

Tachagumpuch, A., *The Anisotropy Ratio of Electrical Conductivity in Graphite*., Submitted in partial fulfillments for the degree of doctor of philosophy at M.I.T. U.S.A. : 1974.

ภาคผนวก

ผู้วิจัยได้เขียนโปรแกรมช่วยคำนวณค่า Q ของตัวสั่นพ้อง โดยใช้ภาษา C ในการโปรแกรม เนื่องจากเป็นภาษาที่ได้รับความนิยมในปัจจุบัน ดังมีรายละเอียดของโปรแกรมดังต่อไปนี้

```
/*Update Verson 5*/
/*True Calibration*/
/***/
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define N 9
#define Pi 3.1415926535

double power(double a,double x){
  return exp(x*log(a));
}

double sum_x(double x[N]){
 double sum=0;
 int i;
 for (i=0;i<N;i++){
   sum+=x[i];
 }
 return sum;
}
```

```
double sum_x2(double x[N]){
 double sum=0;
 int i;
 for (i=0;i<N;i++){
  sum+=(x[i]*x[i]);
 }
 return sum;
}

double sum_x3(double x[N]){
 double sum=0;
 int i;
 for (i=0;i<N;i++){
  sum+=(x[i]*x[i]*x[i]);
 }
 return sum;
}

double sum_x4(double x[N]){
 double sum=0;
 int i;
 for (i=0;i<N;i++){
  sum+=(x[i]*x[i]*x[i]*x[i]);
 }
 return sum;
}

double sum_xy(double x[N],double y[N]){
 double sum=0;
 int i;
```

```
 for (i=0;i<N;i++){
  sum+=(x[i]*y[i]);
 }
 return sum;
}

double mm(double x[N],double y[N]){
 double mmm;
 mmm=((N)*sum_xy(x,y)-sum_x(x)*sum_x(y))/((N)*sum_x2(x)-sum_x(x)*sum_x(x));
 return mmm;
}

double bb(double x[N],double y[N]){
 double bbb;
 bbb=(sum_x(y)*sum_x2(x)-sum_x(x)*sum_xy(x,y))/((N)*sum_x2(x)-sum_x(x)*sum_x(x));
 return bbb;
}

double a_fo(double ri,double rj,double rk,double fi,double fj,double fk){
 double omei=2*Pi*fi;
 double omej=2*Pi*fj;
 double omek=2*Pi*fk;
 double xi=1/(ri*ri-1);
 double xj=1/(rj*rj-1);
 double xk=1/(rk*rk-1);
 double result=0.0;
 double nominator=0.0,denominator=0.0;
 nominator=omei*omei*(xj-xk)+omej*omej*(xk-xi)+omek*omek*(xi-xj);
 denominator=omei*(xj-xk)+omej*(xk-xi)+omek*(xi-xj);
 result=nominator/(2*denominator);
```

```
 return result/(2*Pi);
}

double fo(double ri,double rj,double rk,double fi,double fj,double fk){
 double omei=2*Pi*fi;
 double omej=2*Pi*fj;
 double omek=2*Pi*fk;
 double xi=1/(ri*ri-1);
 double xj=1/(rj*rj-1);
 double xk=1/(rk*rk-1);
 double result;
 double nominator,denominator;
 nominator=omej*omej*(xi-xk)+omei*omei*(xi-xj)-omei*omei*(xi-xk)-omek*omek*(xi-xj);
 denominator=omej*(xi-xk)+omei*(xi-xj)-omei*(xi-xk)-omek*(xi-xj);
 result=nominator/(2*denominator);
 return result/(2*Pi);
}

double ro(double db){
 double x=db/20.0;
 return 1/power(10,x);          /*False */
}

double find_error(double rho[N],double y[N],double ao,double a1){
 double err=0.0;
 double xx[N];
 int j;
 for(j=0;j<N;j++){
   xx[j]=1/(rho[j]*rho[j]-1);
   err+=(xx[j]-((y[j]-ao)/a1))*(xx[j]-((y[j]-ao)/a1));
```

```
/*   err+=(rho[j]-sqrt(fabs(a1/(y[j]-ao))-1))*
       (rho[j]-sqrt(fabs(a1/(y[j]-ao))-1));*/
 }
 return err/N;
}

void main(void){
 double f[N],db[N],rho[N];
 double y1[N],y2[N],y3[N],y4[N],y5[N],y6[N],y7[N],y8[N],y9[N],y10[N];
 double x1[N];
 double ell_over=0.0,ell_under=0.0,psi_under=0.0,psi_over=0.0;
 int i=0,k=0,l=0,mode=N;
 int one=0,oone=0,two=0,three=0 ;
 double fnot=0.0,fnot1=0.0,fnot2=0.0,fokp[5],fokn[5],fnot3=0.0;
 double ao[10],a1[10],error[10],fok[10];
                    d          o          u          b          l          e
q_over=0.0,q_under=0.0,alpha1=0.0,alpha2=0.0,l_over=0.0,l_under=0.0,om_not=0.0;
 for(i=0;i<10;i++){
  ao[i]=0.0;a1[i]=0.0;error[i]=0.0;fok[i]=0.0;
 }
 for(l=0;l<N;l++){
  f[l]=0.0; db[l]=0.0; rho[l]=0.0;
  y1[l]=0.0; y2[l]=0.0; y3[l]=0.0; y4[l]=0.0; y5[l]=0.0; y6[l]=0.0; y7[l]=0.0; y8[l]=0.0;
  y9[l]=0.0; y10[l]=0.0; x1[l]=0.0;
 }

 clrscr();
 gotoxy(5,5);printf("Program Find Q And Fnot");
 gotoxy(5,6);printf("By Tuned Method");
 gotoxy(5,8);printf("Input frequency and dB");
```

```
for(i=0;i<N;i++){
  gotoxy(5,10+i);scanf("%lf %lf",&f[i],&db[i]);
  f[i]=f[i];                    /*scale frequency GHz*/
  rho[i]=ro(db[i]);
}
gotoxy(5,22);printf("Press a key to continue");
getch();
clrscr();
gotoxy(5,5);printf("Your Data");
gotoxy(5,6);printf(" f        dB        rho ");
for(i=0;i<N;i++){
 gotoxy(5,7+i);printf("%e %lf %lf",f[i],db[i],rho[i]);
}
gotoxy(5,22);printf("Press a key to continue");
getch();
clrscr();
/*gotoxy(5,5);printf("Input 3 Index Value Of Data");*/
gotoxy(5,6);printf("To Evaluate Fnot");
/*gotoxy(5,7);scanf("%d %d %d",&oone,&two,&three);*/
if(mode==9){
  fnot1=fo(rho[0],rho[3],rho[6],f[0],f[3],f[6]);
  fnot2=fo(rho[1],rho[4],rho[7],f[1],f[4],f[7]);   /* ????? */
  fnot3=fo(rho[2],rho[5],rho[8],f[2],f[5],f[8]);
  fnot=(fnot1+fnot2)/2;
  }
if(mode==6){
  fnot1=fo(rho[0],rho[2],rho[4],f[0],f[2],f[4]);
  fnot2=fo(rho[1],rho[3],rho[5],f[1],f[3],f[5]);
  fnot=(fnot1+fnot2)/2;
  }
```

```
/*fnot=fo(rho[oone],rho[two],rho[three],f[oone],f[two],f[three]);*/
gotoxy(5,8);printf("%lf %lf %lf %lf",fnot,fnot1,fnot2,fnot3);
/*gotoxy(5,8);printf("%lf",fnot);*/
gotoxy(5,22);printf("Press a key to continue");
getch();
i=1;
for(k=0;k<5;k++){
 fokp[k]=fnot+(double) 0.0002*(double) k;          /*scale GHz*/
 fokn[k]=fnot-(double) 0.0002*(double) i;
 i++;
}
clrscr();
gotoxy(5,5);printf("Frequency around Fnot");
for(k=0;k<5;k++){
 gotoxy(5,6+k);printf("%lf %lf",fokp[k],fokn[k]);
}
gotoxy(5,22);printf("Press a key to continue");
getch();
for(i=0;i<5;i++){
 fok[i]=fokn[4-i];
 fok[i+5]=fokp[i];
}
k=5;l=0;
for(i=0;i<N;i++){
 y1[i]=4*Pi*Pi*(f[i]-fok[l])*(f[i]-fok[l]); x1[i]=1/(rho[i]*rho[i]-1);
 y6[i]=4*Pi*Pi*(f[i]-fok[k])*(f[i]-fok[k]);
}
k=6;l=1;
for(i=0;i<N;i++){
 y2[i]=4*Pi*Pi*(f[i]-fok[l])*(f[i]-fok[l]);
```

```
 y7[i]=4*Pi*Pi*(f[i]-fok[k])*(f[i]-fok[k]);
}
k=7;l=2;
for(i=0;i<N;i++){
 y3[i]=4*Pi*Pi*(f[i]-fok[l])*(f[i]-fok[l]);
 y8[i]=4*Pi*Pi*(f[i]-fok[k])*(f[i]-fok[k]);
}
k=8;l=3;
for(i=0;i<N;i++){
 y4[i]=4*Pi*Pi*(f[i]-fok[l])*(f[i]-fok[l]);
 y9[i]=4*Pi*Pi*(f[i]-fok[k])*(f[i]-fok[k]);
}
k=9;l=4;
for(i=0;i<N;i++){
 y5[i]=4*Pi*Pi*(f[i]-fok[l])*(f[i]-fok[l]);
 y10[i]=4*Pi*Pi*(f[i]-fok[k])*(f[i]-fok[k]);
}
 ao[0]=bb(x1,y1); a1[0]=mm(x1,y1); error[0]=find_error(rho,y1,ao[0],a1[0]);
 ao[1]=bb(x1,y2); a1[1]=mm(x1,y2); error[1]=find_error(rho,y2,ao[1],a1[1]);
 ao[2]=bb(x1,y3); a1[2]=mm(x1,y3); error[2]=find_error(rho,y3,ao[2],a1[2]);
 ao[3]=bb(x1,y4); a1[3]=mm(x1,y4); error[3]=find_error(rho,y4,ao[3],a1[3]);
 ao[4]=bb(x1,y5); a1[4]=mm(x1,y5); error[4]=find_error(rho,y5,ao[4],a1[4]);
 ao[5]=bb(x1,y6); a1[5]=mm(x1,y6); error[5]=find_error(rho,y6,ao[5],a1[5]);
 ao[6]=bb(x1,y7); a1[6]=mm(x1,y7); error[6]=find_error(rho,y7,ao[6],a1[6]);
 ao[7]=bb(x1,y8); a1[7]=mm(x1,y8); error[7]=find_error(rho,y8,ao[7],a1[7]);
 ao[8]=bb(x1,y9); a1[8]=mm(x1,y9); error[8]=find_error(rho,y9,ao[8],a1[8]);
 ao[9]=bb(x1,y10);a1[9]=mm(x1,y10);error[9]=find_error(rho,y10,ao[9],a1[9]);
 clrscr();
 gotoxy(5,5);printf("From Calculation The Results Are..");
 for(k=0;k<10;k++){
```

```
  gotoxy(5,6+k);printf("%d f = %e error = %e",k,fok[k],error[k]);
 }
 gotoxy(5,20);printf("Please choose the Best-fit data number");
 gotoxy(5,21);scanf("%d",&one);
 gotoxy(5,22);printf("Press a key to continue");
 getch();
 alpha2=sqrt((double) fabs(-ao[one]));          /* true */
 alpha1=sqrt((double) fabs(a1[one]-ao[one]));
 om_not=2*Pi*fok[one];
 q_over=om_not/(alpha2-alpha1);
 q_under=om_not/(alpha2+alpha1);
 ell_under=(alpha2-alpha1)/(om_not*om_not);
 ell_over=(alpha2+alpha1)/(om_not*om_not);
 psi_under=om_not/(alpha2-alpha1);
 psi_over=om_not/(alpha2+alpha1);
 clrscr();
 gotoxy(5,5);printf("The final results");
 gotoxy(5,8);printf("fo = %e GHz Q under = %lf Q over = %lf "
                    ,fok[one],q_under,q_over);
 gotoxy(5,10);printf("alpha1 = %e alpha2 = %e",alpha1,alpha2);
 gotoxy(5,12);printf("Psi (under) = %lf  Psi (over) = %lf ",psi_under,psi_over);
 gotoxy(5,14);printf(" ao = %e a1 = %e ",ao[one],a1[one]);
 gotoxy(5,22);printf("Press a key to continue");
 getch();
}
```

ประวัติผู้เขียน

นายจักรพันธ์ ถาวรธิรา เกิดวันที่ 7 มีนาคม พ.ศ.2514 สำเร็จการศึกษาระดับปริญญาตรีวิทยาศาสตรบัณฑิต สาขาฟิสิกส์ ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2535 แล้วเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาฟิสิกส์ ที่ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2536 และได้รับทุนในโครงการผลิตและพัฒนาอาจารย์ของทบวงมหาวิทยาลัยในปีการศึกษา 2539 จบแล้วต้องไปทำงานที่ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา